



INTERNATIONAL CURRICULUM CENTRE

RENDA FUZHONG



IBDP Computer Science Major Assessment 1  
Topic 4: Computational Thinking

Name:

Marks:

Total questions: 6

Total Marks: 55

Exam Time: 80 minutes

Please answer all the questions on the question paper. You are not allowed to use any electronic devices.

- 1 (a) Sheena has inherited a recipe book from her grandmother. All the recipes give ingredient measurements in ounces. Sheena wants to write a program to produce a conversion table that helps her use the correct weight in grams.

To convert ounces into grams: 1 ounce is 28.35 grams.

The conversion table will show the number of grams to the nearest whole number:

Conversion Table	
Ounces	Grams
1	28
2	57
:	:
:	:
16	454

- (i) Sheena writes pseudocode that uses the variables in the table below. Complete the identifier table.

Identifier	Data type	Description
Ounces		Variable used as control variable in FOR loop
Grams		Variable used for storing result of conversion calculation

[2]

- (ii) The built-in function `ROUND(x)` returns `x` rounded to the nearest whole number.

Complete the pseudocode to print the conversion table for 1 to 16 ounces:

```
OUTPUT "    Conversion Table"

OUTPUT "Ounces          Grams"

FOR Ounces ← .....
    Grams ← .....
    ..... ← ROUND(Grams)
    OUTPUT Ounces, "          ", Grams
.....
```

[4]

- (b) Sheena wants to write a function to return the number of boxes of eggs that she needs to buy. The function takes, as a parameter, the number of eggs required for a recipe. There are 6 eggs in a box. Sheena needs to buy enough eggs, but does not want any full boxes of eggs left over. She knows that she can use the operators `DIV` and `MOD` to calculate the required number of boxes.

- (i) Show the results for the following expressions:

```
20 DIV 6 = .....
20 MOD 6 = ..... [2]
```

- (ii) Complete the pseudocode:

```
FUNCTION CalculateNumberOfBoxes (NumberOfEggs : ..... )
    RETURNS .....
    DECLARE ..... : INTEGER
    NumberOfBoxes ← ..... // how many full boxes?
    IF NumberOfEggs MOD ..... // need part of a box?
        THEN ..... // increment number of boxes
        .....
    ENDIF
    RETURN NumberOfBoxes
ENDFUNCTION
```

[5]

2.

Raul copied the following pseudocode from a computing textbook. He wants to find out what it does.

(i) Dry run the pseudocode using the trace table.

```

FOR x ← 2 TO 4
  ThisValue ← List[x]
  y ← x - 1
  WHILE (List[y] > ThisValue) AND (y > 0)
    List[y + 1] ← List[y]
    y ← y - 1
  ENDWHILE
  List[y + 1] ← ThisValue
ENDFOR

```

x	ThisValue	y	List[y]	(List[y] > ThisValue) AND (y > 0)	List			
					[1]	[2]	[3]	[4]
-	-	-	-	-	56	23	67	12
2	23	1	56	TRUE				

[9]

(ii) What does this pseudocode do?

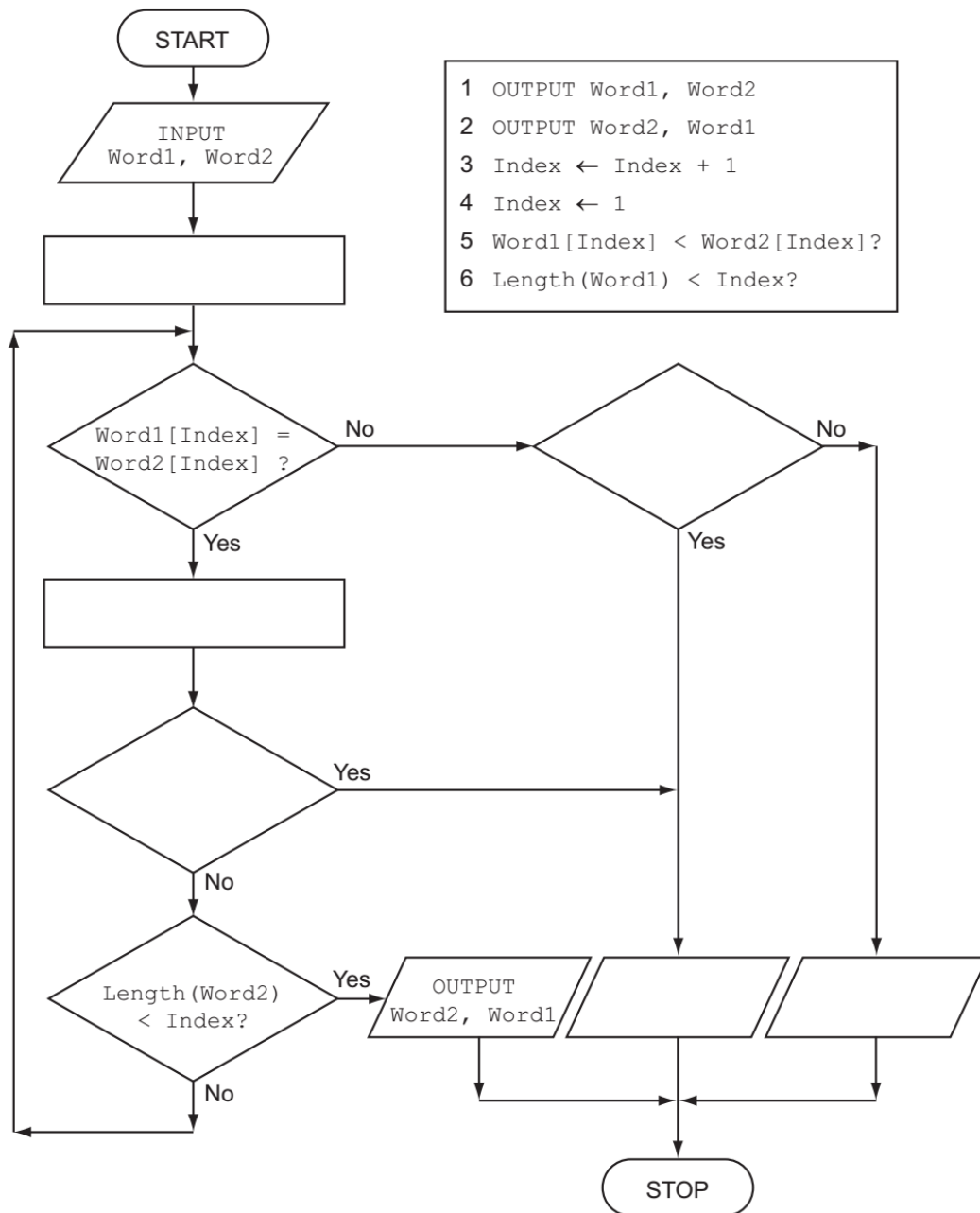
.....

..... [1]

3.

The flowchart outputs the two words in alphabetical order.

Complete the flowchart using the statements in the box below.



[6]

4.

Ashvin is also learning about recursion. He writes the pseudocode for a recursive function.

```
1  FUNCTION Calc(X)
2  DECLARE Temp
3      IF X > 0
4          THEN
5              Temp ← X + Calc(X - 2)
6          ELSE
7              Temp ← 0
8          ENDIF
9  RETURN Temp
10 ENDFUNCTION
```

(a) What is the scope of the variable `Temp`?

..... [1]

(b) State the line number which shows that this function is recursive.

..... [1]

(c) List the function calls that are generated by an initial call `Calc(5)`.

.....  
.....  
.....  
.....  
.....  
..... [3]

(d) What will be returned by the call `Calc(6)`?

..... [1]

5.

Book titles are stored in the file `Book.txt`.

An algorithm is to be designed to perform a serial search of the file for a requested book.

The algorithm will use the variables shown in the table.

(a) Study the table and the algorithm and fill in the gaps.

Identifier	Data Type	Description
<code>Book.txt</code>	FILE	Serial file of book titles
<code>NextBook</code>	STRING	Book title read from the file
<code>IsFound</code>	.....	.....
<code>SearchBook</code>	.....	The requested book

```
//Serial search algorithm
```

```
OPENFILE Book.txt FOR OUTPUT
```

```
INPUT .....
```

```
IsFound ← FALSE
```

```
REPEAT
```

```
    FILEREAD next book data value and assign to NextBook
```

```
    IF ..... = SearchBook
```

```
        THEN
```

```
            IsFound ← TRUE
```

```
            OUTPUT "FOUND"
```

```
        ENDIF
```

```
UNTIL (IsFound = TRUE) OR .....
```

```
IF .....
```

```
    THEN
```

```
        OUTPUT "Book title was NOT FOUND"
```

```
    ENDIF
```

```
..... [8]
```

**(b)** There are 250 book titles in the file.

How many book titles on average will be read to find a requested book title?

.....  
..... [1]

**(c)** The book titles in `Book.txt` are read to the array `BookTitle`.  
A binary search may be an alternative algorithm to a serial search.

**(i)** What condition is put on the `BookTitle` array contents for a binary search to be used?

..... [1]

The following is a recursive function for the binary search algorithm.

```
FUNCTION BinarySearch(ThisArray, FindValue, Low, High) : INTEGER
  IF High < Low
    THEN
      RETURN -1 // not found
    ELSE
      Middle ← INT((High + Low) / 2)
      IF ThisArray[Middle] > FindValue
        THEN
          BinarySearch(ThisArray, FindValue, Low,
                      Middle - 1)
        ELSE
          IF ThisArray[Middle] < FindValue
            THEN
              BinarySearch(ThisArray, FindValue,
                          Middle + 1, High)
            ELSE
              RETURN Middle // found
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDFUNCTION
```

**(ii)** How can you recognise that the function is recursive?

..... [1]

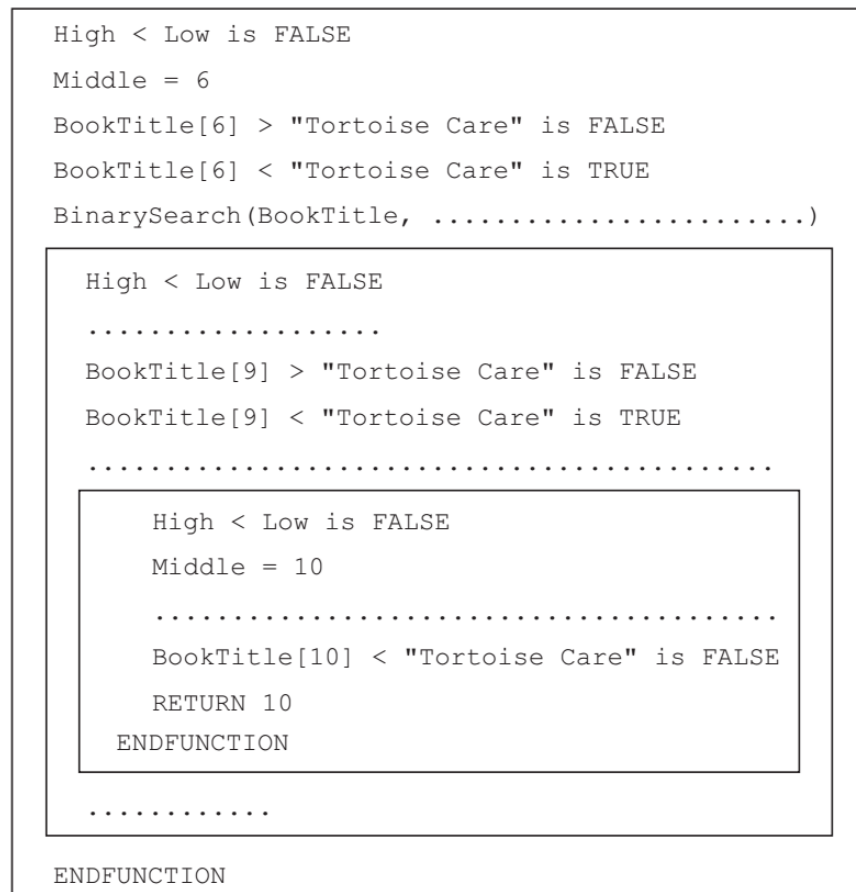
(iii) A binary search is carried out on the following test data in the BookTitle array.

BookTitle	
1	100 Great Artists
2	C++ Made Easy
3	Computing Glossary
4	Database Theory
5	Great Cricket Matches
6	History Of Television
7	Networking
8	Particle Physics
9	String Theory
10	Tortoise Care
11	Visiting China

The trace diagram shown below is for the function call:

BinarySearch(BookTitle, "Tortoise Care", 1, 11)

BinarySearch(BookTitle, "Tortoise Care", 1, 11)



Fill in the gaps in the trace diagram.

[5]



6.

Trace the following algorithmic fragment for  $N = 6$ . Show all working in a trace table.

```
SUM = 0
loop COUNT from 1 to (N div 2)
  if N mod COUNT = 0 then
    SUM = SUM + COUNT
  end if
end loop
if SUM = N then
  output "perfect"
else
  output "not perfect"
end if
```

[4]